APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/747,659	12/26/2003	Pat Styles	13768.783.118	1572

47973   7590   10/31/2007
WORKMAN NYDEGGER/MICROSOFT
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UT 84111

EXAMINER
LEE, MARINA

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
10/31/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
	10/747,659	STYLES ET AL.
Office Action Summary	Examiner	Art Unit	
	Marina Lee	2192	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *26 December 2003*.
2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1-17* is/are pending in the application.
   4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) *1-17* is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.
10) ☒ The drawing(s) filed on *December 26, 2003* is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
   Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
   Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some * c) ☐ None of:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____.
   3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
   * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
   Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413)
   Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

U.S. Patent and Trademark Office
PTOL-326 (Rev. 08-06) Office Action Summary Part of Paper No./Mail Date 20071015

DETAILED ACTION

1.   This action is responsive to the application filed December 26, 2003.

2.   Claims 1-17 are pending and have been examined.

Claim Objections

3.   Claims 10 and 17 are objected to because of the following minor

informalities:

As to claim 10 (lines: 9-10) recites, " a source code files" should be

changed to – "the source code files" –. Appropriate correction is required.

As to claim 17(lines 12) recites, " a binary; and" should be changed to –"a

binary." –. Appropriate correction is required.

Claim Rejections - 35 USC § 101

4.   35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition
> of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the
> conditions and requirements of this title.

5.   Claims 1-10 and 17 are rejected under 35 U.S.C. 101 because the

claimed invention is directed to non-statutory subject matter.

As per claims 1 and 17 recite, "A computer-readable medium having ...".

However, according to Applicant's specification, page 7 "a computer-readable

medium" can be in the form of communication media. A radio frequency and light

wave carrier is only a form of energy that is not a tangible physical article or

object and it does not fall within either of the two definitions of manufacture.

Thus, under the Interim Guidelines such media do not fall within one of the four

statutory classes of 35 U.S.C 101 Annex IV (c). Therefore, the above claims are

non-statutory. For further information, see interim Guidelines for Examination of

Patent Application for Patent Subject Matter Eligibility (Signed 26Oct2005)-OG

Cite: 1300 OG 142. <http://

www.uspto.gov/web/offices/com/sol/og/2005/week47/patgupa.htm>

Claims 2-10 recite the limitation that do not cure the deficiency of the base

claim 1, which regarding to the rejection of non-statutory problem under 35 U.S.C

101. Therefore, they are also rejected for the same reason.

Further regarding to claim 17, recites, " A computer-readable medium

having stored thereon a data structure, comprising: a first data field... a second

data field... a third data field." are presently drafted merely amount to a non-

functional descriptive material as there is no "act" actually being performed. –

See MPEP 2106.01(II).

Claim Rejections - 35 USC § 112

6.   The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly
> claiming the subject matter which the applicant regards as his invention.

7.   Claim 2 is rejected under 35 U.S.C. 112, second paragraph, as being

indefinite for failing to particularly point out and distinctly claim the subject matter

which applicant regards as the invention.

As to claim 2, lines 3 recites the limitation "the debug file". There is

insufficient antecedent basis for this limitation in the claim. Appropriated

correction is required.

Claim Rejections - 35 USC § 103

8.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for

all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described
> as set forth in section 102 of this title, if the differences between the subject matter sought to
> be patented and the prior art are such that the subject matter as a whole would have been
> obvious at the time the invention was made to a person having ordinary skill in the art to which
> said subject matter pertains. Patentability shall not be negatived by the manner in which the
> invention was made.

9.   Claims 1-9 and 11-15 are rejected under 35 U.S.C. 103(a) as being

unpatentable over Grassens, (U.S. Patent No. 7,231,633 B2) and in view of

Haikin, (U.S. Patent No. 6,757,893 B1).

As to claim 1, Grassens discloses a computer readable medium having

computer-executable instructions (e.g., computer program that assist debugging

–*see cols. 6-9: claims 11-19*), comprising:

compiling the source code into executable code (e.g., step 112 – *see col.

2: 62-63*);

obtaining information that identifies the server and the version associated

with the source code (e.g., output file 114, and 500 – see col. 2: 63-64 and col. 4:

58-61);

storing the information in a debug area associated with the executable

code (*see col. 3: 4-7*).

It is noted that Grassens does not specifically discloses storing source

code on <u>a server</u>, the source code associated with a version (e.g., source code

102 and modified source code, --*see Grassens, col. 2: 56-59 and col. 3: 15-23*).

However, Haikin, in an analogous art, teaches a software source code version

control system for use during the development and maintenance of a software

system by multiple software developer in which historical version tracking is

maintained for all source code on a line-by-line basis on central server 20, Fig. 1,

without requiring excessive storage area, in which source code can be accessed

and modified by more than one software developer at a time, in which historical

version tracking of a broad functional changes is provided and in which quick and

transparent access is provided to each version of the source code (see Haikin,

col. 3: 21-31, col. 7: 23-42, and col. 8: 23-59). It would have been obvious to one

ordinary skill in the art at the time the invention was made to manage the change

of the source code 102 of Grassens software source code version control

system, of Haikin for providing flexibility to multiple software developers to work

on the same source code 120 at the same time while still providing historical

version tracking of all modifications to each of the source code lines by storing on

the server storage 270, Fig. 2 (see Haikin col. 1: 15-20).

As to claim 2, Grassens discloses further comprising:

extracting the information from a debug file (see col. 3: 31-41 and col. 4:

58-65);

requesting the source code associated with the version from the server via

the information (see col. 4: 62-65) ;

placing the source code in a directory used by a debugger to debug the

executable code (see col. 3: 14-20 and col. col. 4: 54-59);

executing the debugger and matching an instruction in the executable code to an instruction in the source code (see col. 3: 7-10 and col. 4: 59-67 and col. 5: 1-16).

As to claim 3, Grassens further discloses wherein the source code includes programming statements which, wherein compiled, produce executable code (e.g., printf statement of source code 300, Fig. 3 – see co. 4: 7-12 & 51-53).

As to claim 4, Haikin further discloses wherein the server comprises a version control server that stores a plurality of versions of the source code (e.g., version control server module 280, Fig. 2, see Haikin, col. 8: 38-42).

As to claim 5, modified Grassens with Haikin further discloses wherein the information comprises a name of the server, a port of the server at which the server may be accessed to access the source code (e.g., server 20, Fig. 1 – see Haikin, col. 7: 20-42), a path to the source code (e.g., source file name 502, Fig. 5 – see Grassens, col. 4: 58-61), and a numeric value that indicates a version number (e.g., version 1.0 of the version code 502, Fig. 5 – see Haikin, col. 10: 26-30) of the source code.

As to claim 6, Grassens further discloses wherein the executable code includes code that was compiled from a plurality of source code files, each source code file associated with a version (e.g., source code 102 and modified source code, --see col. 2: 56-59 and col. 3: 15-23).

As to claim 7, Grassens discloses further comprising obtaining additional information that identifies the versions associated with the plurality of source

code files to the server and storing the additional information in the debug file

(e.g., output file 114, and 500 – *see col. 2: 63-64, col. 3: 4-7, and col. 4: 58-61*).

As to claim 8, Grassens further discloses wherein the debug area

comprises a program database file that is separate from eh executable code

(e.g., explicitly program output 114 and execution log 116, Fig 1, -- *see col. 2: 63-*

64).

As to claim 9, Grassens further discloses wherein the debug area

comprises a portion of an executable file that includes the executable code (e.g.

low level verbosity – *see col. 3: 34-36*).

As to claim 11, Grassens discloses a system for debugging binaries,

comprising:

a compiler arranged to compile source code files (e.g., source code 102

and modified source code, --*see Grassens, col. 2: 56-59 and col. 3: 15-23*) into a

binary (e.g., execution 116, Fig. 1, -- *see col. 2: 64*) and to generate debug data

(e.g., step 112 – *see col. 2: 62-63)*;

an extractor arranged to determine information(see col. 2: 63-64 and col.

4: 58-61) including the version of each source code file (see col. 4: 62-65) used

to create the binary and store the information for use in retrieving the source

code files at a debug time (see col. 3: 31-41 and col. 4: 58-65).

It noted that Grassens does not explicitly disclose a version control server

arranged to store versions of the source code. However, Haikin, in an analogous

art, teaches a software source code version control system for use during the

development and maintenance of a software system by multiple software

developer in which historical version tracking is maintained for all source code on a line-by-line basis on central server 20, Fig. 1, without requiring excessive storage area, in which source code can be accessed and modified by more than one software developer at a time, in which historical version tracking of a broad functional changes is provided and in which quick and transparent access is provided to each version of the source code (see Haikin, col. 3: 21-31, col. 7: 23-42, and col. 8: 23-59). It would have been obvious to one ordinary skill in the art at the time the invention was made to manage the change of the source code 102 of Grassens software source code version control system, of Haikin for providing flexibility to multiple software developers to work on the same source code 120 at the same time while still providing historical version tracking of all modifications to each of the source code lines by storing on the server storage 270, Fig. 2 (see Haikin col. 1: 15-20).

As to claim 12, Grassens discloses further comprising a source server arranged to extract the information at debug time, retrieve the source code files from the version control server, and place the source code files in the directory accessible by a debugger (see col. 3: 14-20 and col. col. 4: 54-59).

As to claim 13, Grassens discloses wherein the source server comprises a component of the debugger (see col. 3: 44-47).

As to claim 14, Grassens discloses wherein the source server is separate from the debugger (see col. 3: 44-47).

As to claim 15, Grassens discloses wherein the debugger is arranged to fine the source code files in the directory and is unaware of the version control server (see col. 3: 32-41 and col. 4: 58-61).

10.   Claims 10 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Grassens, (U.S. Patent No. 7,231,633 B2) and in view of Haikin, (U.S. Patent No. 6,757,893 B1), as applied to claims: 1-9 and 11-15 above respectively, and in further view of Schmidt et al., (hereinafter – Schmidt), (U.S. Patent No. 4,558,413).

As to claim 10, modified Grassens with Haikin discloses further comprising:

iterating each source code file that is part of a compilation (see col. 3: 14-23), each source code file having a version(e.g., source code 102 and modified source code,--see col. 2: 56-59 and col. 3: 15-23) ;

obtaining information that identifies the version of each source code file to the server and a local name of each source code file (e.g., output file 114, and 500 – see col. 2: 63-64 and col. 4: 58-61);

extracting, from a binary, local names of a source code files that were used in compiling the binary (e.g., source code file name 502, Fig. 5 -- see Grassens, col. 4: 58-61).

It is noted that Grassens does not explicitly disclose storing the information in a lookup table; and for each source code file that was used in compiling the binary, look up the version in the lookup table by using the local name of the source code file. However, Schmidt, in an analogous art, teaches

Object type table that contains a list of object that are referenced by models and

haven been analyzed as to their type, e.g., the modeler abstracts essential

properties of the objects in a models and stores the information in this table, e.g.

a cedar source file is listed along with the implied procedure type used by the

modeler to compile and load it. The unique name of an object is the key in this

table and its type is the value. The object type table also contains information

that records whether a file has been edited and if so, whether it has been saved

on a remote file server (see Schmidt, col. 50: 16-27).

It would have been obvious to one ordinary skill in the art at the time the

invention was made to store the source code 102 and the modified source code

of Grassens with Haikin in the Project Table of Schmidt as a way to speed up the

analysis of source code file 102 (see Schmidt, col. 51: 32-33).

As to claim 16, it is noted that modified Grassens with Haikin does not

specifically disclose further discloses wherein the information comprises a name

of the sever, a port of the server (e.g., server 20, Fig. 1 – see Haikin, col. 7: 20-

42), a path to the source code (e.g., source file name 502, Fig. 5 – see

Grassens, col. 4: 58-61), and a numeric value that indicates a version number

(e.g., version 1.0 of the version code 502, Fig. 5 – see Haikin, col. 10: 26-30) of

the source code.

It is noted that modified Grassens with Haikin does not specifically

disclose further discloses wherein the information comprises key values of the

control server indicating at which the version control server may be accessed to

access the source code. However, Schmidt, in an analogous art, teaches Object

type table that contains a list of object that are referenced by models and haven

been analyzed as to their type, e.g., the modeler abstracts essential properties of

the objects in a models and stores the information in this table, e.g. a cedar

source file is listed along with the implied procedure type used by the modeler to

compile and load it. The unique name of an object is the key in this table and its

type is the value. The object type table also contains information that records

whether a file has been edited and if so, whether it has been saved on a remote

file server (see Schmidt, col. 50: 16-27). It would have been obvious to one

ordinary skill in the art at the time the invention was made to store the source

code 102 and the modified source code of Grassens with Haikin in the Project

Table of Schmidt as a way to speed up the analysis of source code file 102 (see

Schmidt, col. 51: 32-33).

11.   Claims 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Haikin, (U.S. Patent No. 6,757,893 B1) and in view of Grassens, (U.S. Patent No.

7,231,633 B2).

        As to claim 17, Haikin discloses a computer-readable medium (e.g., server

disk 25 – *see col. 8: 24-45*) having stored thereon a data structure, comprising:

        a first data filed containing data that identifies a version control server that

stores a plurality of source code files(e.g., source code storage 270 and 401 –

see col. 8:35-38 and col. 9: 32-42), each source code file associated with at least

one version (e.g., version control server module 280, Fig. 2, -- *see col. 38-42*);

a second data field containing data that identifies a port of the version

control server at which the source code field are accessed (e.g., user code 503,

Fig. 5 –"all user" or "specific user name: BOB" – see col. 10: 40-54);

It is noted that Haikin does not explicitly disclose a third data field

containing data that identifies a set of source code files including version

information compiled to create a binary. However, Grassens, in an analogous art,

teaches placing output statements at periodic intervals thorough out the source

code of the computer program. The computer program is analyzed for proper

locations for output statements which may have different levels of verbosity that

are controllable within the source code and by the debugging program (see

Haikin, Title, Abstract, col. 1: 40-47, and col. 3: 4-11).

It would have been obvious to one ordinary skill in the art at the time the

invention was made to include the source code storage 270 of Haikin with the

debugging trace output statement 114 and executable log file 116 of Grassens

for enable a tester or developer to efficiently trace the flow of the program and

help in debugging program once applied in Grassens (see Grassens col. 1: 30-

35).

Conclusion

12.   The prior art made of record and not relied upon is considered pertinent to

applicant disclosure.

Bates et al. (US 6,964,036 B2) is cited to teach descriptive variables while

debugging.

13.   Any inquiry concerning this communication or earlier communications from
the examiner should be directed to Marina Lee whose telephone number is (571)
270-1648. The examiner can normally be reached on M-F (11:00 am to 7: 30
pm) Est.

If attempts to reach the examiner by telephone are unsuccessful, the
examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The
fax phone number for the organization where this application or proceeding is
assigned is 571-273-8300.

Information regarding the status of an application may be obtained from
the Patent Application Information Retrieval (PAIR) system. Status information
for published applications may be obtained from either Private PAIR or Public
PAIR. Status information for unpublished applications is available through
Private PAIR only. For more information about the PAIR system, see http://pair-
direct.uspto.gov. Should you have questions on access to the Private PAIR
system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-
free). If you would like assistance from a USPTO Customer Service
Representative or access to the automated information system, call 800-786-
9199 (IN USA OR CANADA) or 571-272-1000.

M.L.

October 15, 2007

TUAN DAM
SUPERVISORY PATENT EXAMINER